# ICASE

ON CONSERVATION AT GRID INTERFACES

Marsha J. Berger

Report No. 84-43

September 7, 1984

INSTITUTE FOR COMPUTER APPLICATIONS IN SCIENCE AND ENGINEERING
NASA Langley Research Center, Hampton, Virginia 23665

Operated by the

UNIVERSITIES SPACE RESEARCH ASSOCIATION

# On Conservation at Grid Interfaces

*Marsha J. Berger**

Courant Institute of Mathematical Sciences
251 Mercer St.
New York University
New York, NY 10012

*ABSTRACT*

This paper considers the solution of hyperbolic systems of conservation laws on discontinuous grids. In particular, we consider what happens to conservation at grid interfaces. A procedure is presented to derive conservative difference approximations at the grid interfaces for two dimensional grids which overlap in an arbitrary configuration. The same procedures are applied to compute interface formulas for grids which are refined in space and/or time, and for continuous grids where a switch in the scheme causes the discontinuity.

*Supported in part by the National Science Foundation under Grant No. MCS77-02082, by Department of Energy Contract No. DEAC0276ER03077-V, and by the National Aeronautics and Space Administration under NASA Contract No. NAS1-17070, while the author was in residence at the Institute for Computer Applications in Science and Engineering, NASA Langley Research Center, Hampton, VA 23665.

## 1. Introduction

This paper considers the solution of hyperbolic systems of conservation laws on discontinuous grids. The discontinuity in the grid can be due to the overlapping of two different mesh systems, or to one grid which undergoes an abrupt mesh refinement in space and/or time. Some of our results include the case where the discontinuity is due to switching numerical schemes on the same grid, rather than the switching of the grids themselves. In either case, since the solution is discontinuous, some analog to conservative differencing is needed at the grid interface. Following Lax and Wendroff [1960] we give interface conditions that guarantee that if the solution converges, it converges to a weak solution of the equations, and thus insures correct jump conditions for shocks.

Computations with internal interfaces are becoming more common from two sources. First, many large scale scientific computations, for example in transonic flow, are modeling more and more complex geometries in two and three space dimensions. As the configurations that can be modeled become more complex, so does the grid generation problem. It is very difficult to generate one smooth body-fitted coordinate transformation around a general possibly multiply connected domain. To simplify this problem, it is becoming more common to use several grids at once, each in a different coordinate system (Boppe, [1980]; Atta, [1981]; Benek, Steger and Dougherty, [1983]). Each part of the domain will still have a body fitted grid, but now the component grids will in general overlap each other in an irregular fashion, (for example see figure 1.1). Thus, additional boundary conditions will be needed for those grids with a boundary that is interior to the problem domain.

The second source of these interface problems comes from the use of adaptive techniques, where a grid is abruptly refined in space and time. In this approach, used by (Bolstad [1982]; Gropp [1980]; Berger and Oliger [1984]; Berger and Jameson [1983]), rectangular fine grids are superimposed on an underlying coarse grid in those regions where the solution accuracy is inadequate. Thus, this can be regarded as a special case of the independent grids described above. In addition to being refined in space, the grids are refined in time as well, so that smaller steps are taken on the fine grids. This is done so that the mesh ratio $\lambda_{fine} = \lambda_{coarse}$, so the same integrator is stable on each grid. The small timestep does not have to be applied on the entire grid. (See figure 2.2 for an illustration of this.) Similar ideas have been proposed in the multigrid literature, for example see Brandt [1981] and McCormick [1984].

Both multiple component grids and mesh refinement will be even more necessary in 3D calculations. For difficult problems involving for example wings, nacelles, engines,

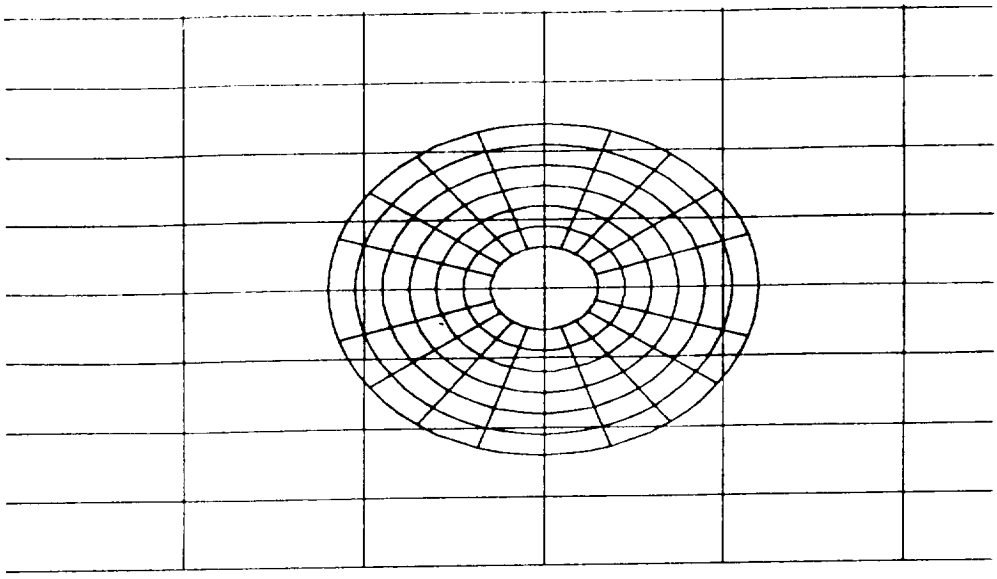

**Figure 1.1** Independently generated component grids for flow in a channel around an obstacle.

canards, etc., if one insists on using one global coordinate system for the grid discretization, we believe that some type of local refinement will be necessary to make up for the extreme grid skewness and uneven resolution that these global grids can't avoid. Thus, there will still be an interface problem.

In both of the cases described above, it is important to use conservative interface procedures to guarantee the correct shock location for shocks passing through grid boundaries, and to insure that artificial shocks are not generated at grid interfaces. We give a general procedure for deriving conservative interface conditions that give weak solutions to the differential equation if they converge. The method gives formulaes of arbitrary order of accuracy, but the higher order ones may be unstable. In some cases, the resulting formulas are not new, although the derivation is. In other cases, the different approach leads to new formulas, for example in the case of overlapping grid systems, which are not immediately equivalent to a finite volume approach for determining fluxes across the boundary. Some of our results are still preliminary, in that numerical experiments have not yet been done.

In the rest of this section we review some basic facts about conservation and set notation for the rest of the paper. Our basic approach follows from a precise definition of conservation on irregular grids (by which we mean overlapping component grids or grids with an abrupt refinement in space or time). In section 2 we use this approach to derive

interface conditions for an interface due to an abrupt refinement in space, then space and time. In section 3, we formalize the approach illustrated in section 2, and discuuss the order of accuracy. In section 4 we look at the case of overlapping meshes in one space dimension, as a prelude to the more difficult problems in two dimensions in section 5. We present an algorithm for determining conservative interface conditions for grids which intersect in an arbitrary way in two dimensions. However, in the general case, there are many open questions remaining which will be important for numerical computations using independently generated component grids.

Consider the solution of a hyperbolic system of conservation laws

$$u_t + f(u)_x = 0, \tag{1.1}$$
$$u(x,t=0) = u_0(x).$$

In cases where the solution is discontinuous, a classical solution by definition no longer exists. Instead, $u$ is a weak solution if

$$\int\int u\phi_t + f(u)\phi_x \, dx \, dt + \int u_0(x)\phi(x) \, dx = 0, \tag{1.2}$$

for any smooth test function $\phi(x,t)$ with compact support. In problems with a discontinuity in the solution, the speed $s$ of the discontinuity is given by the Rankine Hugoniot condition,

$$s[u] = [f], \tag{1.3}$$

where the brackets denote the jump in the quantity at the discontinuity. This condition follows directly from (1.2), see Lax [1972]. It is well known that the numerical computation of a discontinuous solution of (1.1) requires the numerical scheme to be in conservation form. An explicit finite difference approximation to (1.1) is in conservation form if it can be written

$$\begin{aligned} v_j^{n+1} &= Q(v_{j-p}^n, \cdots, v_{j+q+1}^n) \\ &= v_j^n - \frac{\Delta t}{\Delta x} \Delta_+ g(v_{j-p}^n, \ldots, v_{j+q}^n), \end{aligned} \tag{1.4}$$

where $g$ is a Lipschitz continuous numerical flux function and $\Delta_+$ is the forward difference operator. Consistency requires that $g(v,v,\ldots,v) = f(v)$.

The question of conservation when switching between two different grids or numerical schemes has been considered by several authors. Warming and Beam [1976] derived transition operators for switching conservatively between MacCormack's method and a second order upwind scheme. Hessenius and Pulliam [1982] applied this transition operator approach to derive so-called zonal interface conditions, however, with a

significant loss of accuracy at the zonal interfaces. Rai [1984] has developed conservative zonal interface conditions for zonal grids which share a common grid line, and has beautiful calculations demonstrating the shock capturing ability of zonal grids with a discontinuity crossing zones. The derivation presented here contains his algorithm as a special case. The need for conservative grid interfaces is illustrated by Benek, Steger, and Dougherty [1983]. Their transonic flow calculations use grids in two coordinate systems, one around an airfoil and one smaller embedded grid around a flap. The computed shock is very distorted as it passes through the grid interface.

Finally, Osher and Sanders [1984] have conservative interface equations for a mesh which is refined by an integer $r$ in time. However, their scheme requires the savings of $r$ intermediate values in time at the boundary of the fine grid. Our approach (or a variant of theirs) yields conditions without this drawback, and so are easier to implement.

The approach we take is based upon the direct numerical approximation of (1.2), and the proof that a convergent conservation form scheme converges to a weak solution. We repeat the main argument of the Lax and Wendroff proof to set the stage for the further development, which also uses summation by parts. Take the conservation form scheme (1.4), multiply by the test function $\phi_j^n$, $\Delta x$, and $\Delta t$, and sum over all grid points $j$, and $n \geq 0$, giving

$$\sum_j \sum_{n\geq 0} \frac{v_j^{n+1} - v_j^n}{\Delta t} \phi_j^n \, \Delta t \Delta x \tag{1.5}$$

$$= -\sum_j \sum_{n\geq 0} \frac{g(v_{j-p+1}, v_{j+q+1}) - g(v_{j-p}, v_{j+q})}{\Delta x} \phi_j^n \, \Delta t \Delta x.$$

Applying summation by parts to the left sidde of (1.5) gives

$$\sum_j \sum_{n\geq 0} \frac{v_j^{n+1} - v_j^n}{\Delta t} \phi_j^n \Delta t \Delta x = \sum_j \sum_{n\geq 1} \frac{v_j^n (\phi_j^{n-1} - \phi_j^n)}{\Delta t} \Delta t \Delta x - \sum_j v_j^0 \phi_j^0 \Delta x,$$

which converges to $\int\int -u\phi_t dx\, dt - \int u_0(x)\phi(x)dx$ under suitable hypotheses as $\Delta x, \Delta t \to 0$. A similar summation is done on the right hand side of (1.5).

Another way to say the above is that the integral in definition (1.2) is being approximated by the trapezoid rule,

$$\int u \, dx = \sum_j u_j \, h + O(h^2). \tag{1.6}$$

Instead of exactly conserving the quantity $I(t) = \int u(x,t)dx$, except for the flux at the boundaries of the domain, the discrete approximation

$$S^n = \sum h \, u_j^n \tag{1.7}$$

is conserved exactly by the numerical scheme. It is this integral approximation (1.6) and numerically conserved quantity (1.7) that we generalize when considering conservation on irregular grids.

## 2 Interfaces Due to an Abrupt Mesh Refinement

To develop the procedure, we first consider the very simple case of an abrupt mesh refinement in space. We will show how to derive a class of conservative interface conditions that give weak solutions to the differential equation if they converge.

For ease of presentation, we consider the case of mesh refinement by a factor of 2. Assume the interface between the coarse and fine grids is located at $x = 0$, where we use the notation of figure 2.1.

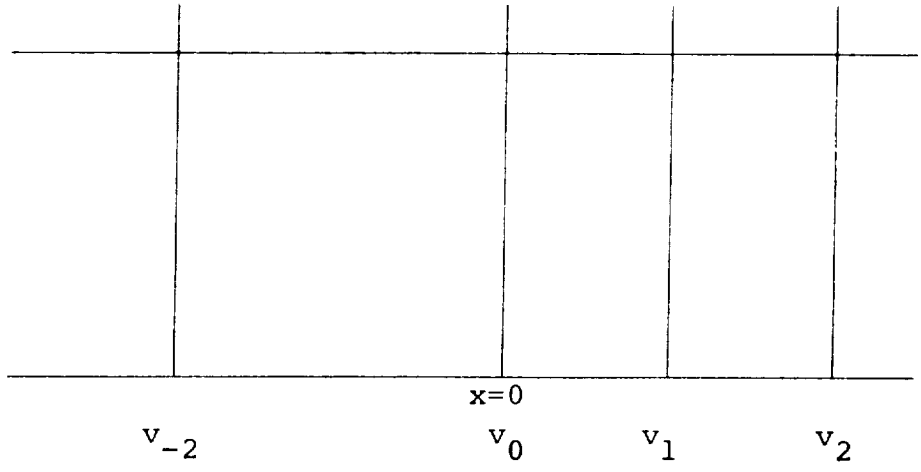


**Figure 2.1** Interface for mesh refinement in space only.

Assume on either side of the interface we integrate using a scheme in conservation form with a 3 point stencil. The scheme is

$$\frac{v(x,t+k)-v(x,t)}{k} = \begin{cases} \dfrac{g(x,x+h)-g(x-h,x)}{h} & x \geq h \\ \dfrac{g(x,x+2h)-g(x-2h,x)}{2h} & x \leq -2h \\ Q(-2h,0,h) & x=0 \end{cases} \tag{2.1}$$

The goal is to determine the unknown function $Q$ for the interface equation for $v(x=0,t)$ in such a way that conservation is maintained.

Following Lax and Wendroff, we multiply (2.1) by a smooth test function $\phi(x,t)$ and the mesh widths in time and space. Summation by parts will again put the differences on $\phi$, which should converge to the derivatives of $\phi$, and the sums to integrals in space and time. However, in this case there will be boundary terms arising from the summation by parts at the interface, where the mesh width changes. $Q$ can then be determined by requiring that these boundary terms all vanish up to terms of order $h$. In addition, there is some flexibility as to which discrete integral approximation to use, although any integration formula will give a consistent boundary approximation. For the summation in space, for example, we can consider the one-sided approximation to the integral

$$\int_{-\infty}^{\infty} v(x,t)dx \approx \sum_{x\le -2h} v(x,t)2h + \sum_{x\ge 0} v(x,t)h, \tag{2.2}$$

or the more accurate trapezoid rule,

$$\int_{-\infty}^{\infty} v(x,t)dx \approx \sum_{x\le -2h} v(x,t)2h + \sum_{x\ge h} v(x,t)h + \frac{3h}{2}v(0,t). \tag{2.3}$$

If the interface formula $Q$ is derived using (2.2), then the solution will exactly conserve at each time step the discrete approximation on the right hand side of (2.2), and similarly if (2.3) is used.

Suppose we multiply (2.1) by $\phi$, and use the rule (2.2) to form the sums. The interesting thing is what happens to the right hand side, which is

$$\sum_{x\le -2h}\left[\frac{g(x,x+2h)-g(x-2h,x)}{2h}\right]\phi(x,t)2hk + \tag{2.4}$$

$$\sum_{x\ge h}\left[\frac{g(x,x+h)-g(x-h,x)}{h}\right]\phi(x,t)hk$$

$$+ Q(-2h,0,h)\phi(0,t)hk =$$

$$\sum_{x\le -2h} g(x,x+2h)\left[\frac{\phi(x,t)-\phi(x+2h,t)}{2h}\right]2hk + g(-2h,0)\phi(-2h,t) \tag{2.5}$$

$$+ \sum_{x\ge h} g(x,x+h)\left[\frac{\phi(x,t)-\phi(x+h,t)}{h}\right]hk - g(0,h)\phi(h,t)$$

$$+ Q(-2h,0,h)\phi(0,t)hk.$$

The two sums in (2.5) already form an $O(h)$ approximation to the integral in space. We must have that the three terms remaining cancel up to terms of order $h$, or

$$hQ(-2h,0,h)\phi(0,t) + g(-2h,0)\phi(-2h,t) - g(0,h)\phi(h,t) = O(h). \tag{2.6}$$

Expanding the test function $\phi(x,t)$ around $x=0$ gives

$$\Big[hQ + g(-2h,0) - g(0,h)\Big]\phi(0,t) = O(h). \tag{2.7}$$

The interface condition $Q$ is therefore determined by requiring

$$hQ + g(-2h,0) - g(0,h) = 0. \tag{2.8}$$

For example, using the numerical flux function $g$ for the Law-Wendroff difference scheme,

$$g(v_{j-1},v_j) = \frac{f(v_{j-1})+f(v_j)}{2} + \frac{\lambda}{2}A\left(\frac{v_{j-1}+v_j}{2}\right)(f(v_j)-f(v_{j-1})),$$

where the Jacobian $A = \dfrac{\partial f}{\partial u}$, the resulting interface approximation is

$$v(0,t+k) = v(0,t) + \frac{\lambda}{2}(f(h)-f(-2h)) \tag{2.9}$$
$$+ \frac{\lambda^2}{2}\left[A(\frac{h}{2})(f(h)-f(0))-2A(-h)(f(0)-f(-2h))\right].$$

This is not a very accurate approximation, however. For smooth solutions the local truncation error is only $O(h)$. However, since the scheme is only applied at one point we can still get convergence (in the linear case) if the combined scheme is GKS-stable, since the method applied everywhere else is at least $O(kh)$ (Gustafsson, [1975]).

If however, the more accurate integral approximation (2.3) is used, the same procedure yields the interface approximation

$$\frac{3h}{2}\,Q(-2h,0,h) = g(0,h) - g(-2h,0). \tag{2.10}$$

For the Lax-Wendroff scheme this gives the interface equation

$$v(0,t+k) = v(0,t) + \frac{\lambda}{3}(f(h)-f(-2h)) \tag{2.11}$$
$$+ \frac{\lambda^2}{3}\left[A\left(\frac{h}{2}\right)(f(h)-f(0))-2A\left(-h\right)\right].$$

This is a more accurate approximation, with local truncation error $O(kh)$. Either of these interface conditions could have been easily derived using a finite volume approach, and balancing the flux at the interface. However, this formalism will help in more complicated situations.

For the case of mesh refinement in time and space, we again consider the simplest case of refinement by a factor of 2, using the notation of figure 2.2. Interface equations are needed for $v_0^{1/2}$ and $v_0^1$ so that global conservation is maintained.

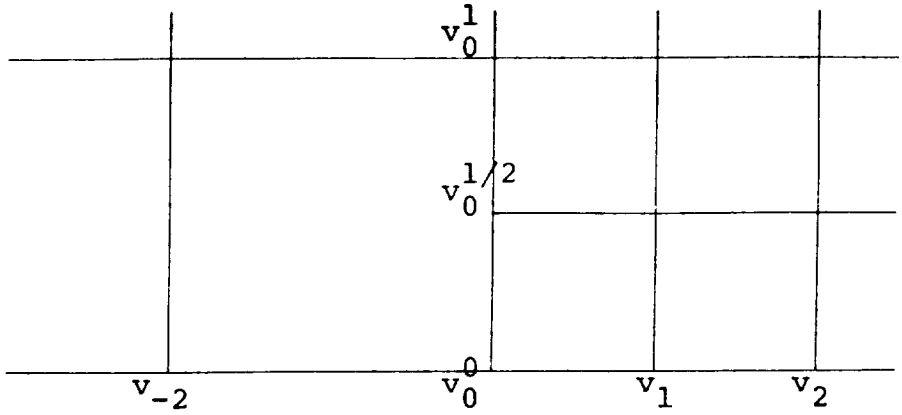


**Figure 2.2** Interface for mesh refinement in space and time.

Again assuming a three point stencil, the scheme is

$$\begin{aligned}
\frac{v(x,t+k)-v(x,t)}{2k} &= \frac{g(x,x+2h)-g(x-2h,x)}{2h} \qquad x<0 \\
\frac{v(x,t+k)-v(x,t)}{k} &= \frac{g(x,x+h)-g(x-h,x)}{h} \qquad x>0 \\
\frac{v(0,t+k)-v(0,t)}{k} &= Q \\
\frac{v(0,t+2k)-v(0,t+k)}{k} &= R
\end{aligned} \tag{2.15}$$

Again, all equations in (2.15) are multiplied by $\phi(x,t)$ and either $2h$ $2k$ for $j<0$, $hk$ for $j>0$, or $\frac{3h}{2}k$ for $j=0$, using the trapezoid rule. When summation by parts is applied to the terms with the differencing in space, each sum will produce a boundary term. The requirement that the boundary terms vanish to order $h$ gives (dropping the summation in time)

$$2g^n_{-1/2}\phi^n_{-1} - g^n_{1/2}\phi^n_1 - g^{n+1/2}_{1/2}\phi^{n+}_1 + R\phi^{n+1/2}_0\frac{3h}{2} + Q\phi^n_0\frac{3h}{2} = O(h).$$

This can be accomplished by defining

$$R = \frac{g^{n+1/2}_{1/2} - g^n_{-1/2}}{\frac{3h}{2}}, \tag{2.16}$$

and

$$Q = \frac{g^n_{1/2} - g^n_{-1/2}}{\frac{3h}{2}},$$

These interface conditions can be shown to be first order accurate, with local truncation error $O(kh)$. For mesh refinement in time and space by an arbitrary integer $r$, the formulas generalize immediately, and have the interesting stencil shown in figure 2.3.

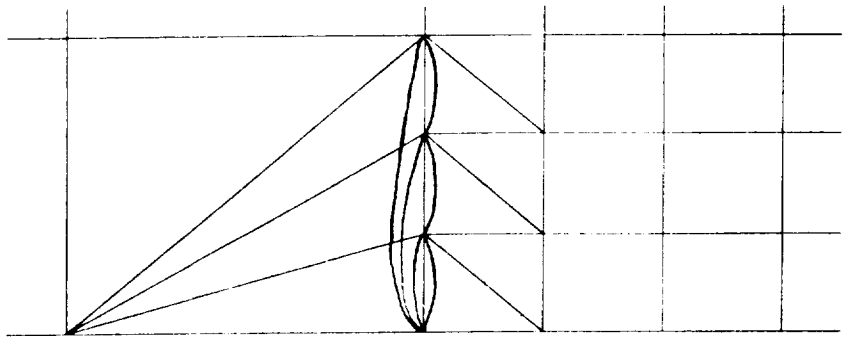

**Figure 2.3** Interface stencil for mesh refinement in time and space.

It is difficult to prove stability of the interface equations (2.16). For mesh refinement in space only, the conservative interface conditions (2.11) using Lax-Wendroff to advance the fine and coarse grids can be shown to be stable in the sense of Gustaffson, Kreiss and Sundstrom [1972] (Berger, [1982]). In numerical experiments, the interface equations for mesh refinement in time and space are stable, and give good results when either a discontinuity or rarefaction wave passes through the interface.

Unfortunately, in the non-dissipative case, more care is required to develop stable and conservative interface equations. Consider a mesh refined in space only by a factor of 2, and use cell-centered variables as indicated in figure 2.4. We will consider the semi-discrete version of $u_t = u_x$ with central differencing on either side of the interface. One possibility for conservation across the line $x = 0$ is given by the equations

$$u_{jt} = \frac{u_{j+1} - u_{j-1}}{4h} \tag{2.17}$$

$$= \frac{1}{2h}\left[\frac{(u_{j+1}+u_j)}{2} - \frac{(u_j+u_{j-1})}{2}\right]$$

$$v_{jt} = \frac{v_{j+1} - v_{j-1}}{2h} \qquad j>1$$

$$v_{0t} = \frac{1}{h}\left[\frac{(v_0+v_1)}{2} - \frac{(u_0+u_1)}{2}\right] = \frac{v_0+v_1-u_0-u_1}{2h}$$

In (2.17), we use $u_1 = \frac{v_0+v_1}{2}$. This set of equations supports a standing oscillation on the fine grid, emanating from the interface, with no such wave on the coarse grid. Thus, energy is radiating from the interface without any incident waves, and so the interface given is unstable.

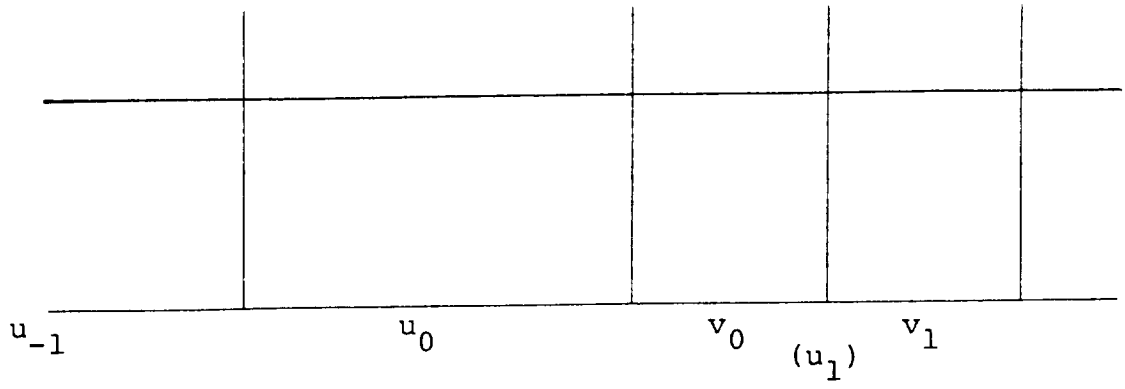


**Figure 2.4** Mesh refinement in space with cell-centered variables.

An alternate approach is to interpolate for $v_{-1}$ using $u_0$ and $u_1$ of the coarse grid, and use the interface equations

$$v_{0t} = \frac{v_1 - v_{-1}}{2h} \tag{2.18}$$

$$u_{0t} = \frac{1}{2h}\left[\frac{(v_{-1}+v_0)}{2} - \frac{(u_{-1}+u_0)}{2}\right]$$

This interface condition does not excite the unstable oscillatory mode in (2.17). The two dimensional nonlinear analog of equation (2.18) has been tested by computing steady transonic flow with the Euler equations, (Berger and Jameson, [1983]).

### 3 Discrete Approximations to Integrals

In this section we formalize the approach illustrated in the examples in section 2. We assume a computation is done using one grid, with an interface at the point $x = 0$. The interface can be due to a mesh refinement, or to using one numerical scheme on the

left, and a different one on the right. A class of conservative interface formulas will be derived at the point $x = 0$.

The goal is to obtain a weak solution of (1.2). By approximating the integral using a general quadrature formula, the interface equation will again be determined so that the discrete formulas is exactly conserved at each time step. To do this, we first quote the following lemma from (Mock and Lax, [1978]). They state

*Lemma: Let $f$ be any $C^\infty$ function on $R_+$ with bounded support. Given any positive integer $\nu$, there exists a quadrature formula accurate to order $\nu$ of the form*

$$\int_0^\infty f(x)\,dx = h\sum_0^\infty w_j f(jh) + O(h^\nu),$$

*where the weights depend on $\nu$, but $w_j = 1$ for $j \geq \nu$ .*

The weights $w_j$ are derived from the Euler MacLaurin summation formula. An integration rule for the whole domain is derived by putting two one-sided formulas together. Since the weights are constant equal to 1 in the interior of the grid, a scheme in conservation form will conserve this accurate approximate except at boundary regions. In what follows, we will ignore all boundary terms except those arising from the internal interface.

We can thus write

$$\int u\,dx = \sum_j h\,w_j\,u_j + O(h^\nu), \tag{3.1}$$

and define

$$S^n = \sum_j h\,w_j\,u_j^n. \tag{3.2}$$

To exactly conserve $S$ at each step, we have

$$\begin{aligned} S^1 &= h\sum_j w_j u_j^1 \\ &= h\sum_{j\neq 0} w_j\,(u_j^0 + \frac{\Delta t}{h}(g(u_j,u_{j+1}) - g(u_{j-1},u_j)) + (u_0^0 + \Delta t Q u_0^0)w_0 h\,, \end{aligned}$$

where for ease of notation we assume a two point flux function $g$. Clearly $S^1 = S^0$ if

$$Qu_0 = \frac{\sum_{j=-\nu}^{\nu} g(u_j,u_{j+1})(w_{j+1}-w_j) + w_0(g_+ - g_-)}{w_0 h} \tag{3.3}$$

where we use $w_{|j|} = 1$ for $|j| \geq \nu$. Note that this approximation can have a big stencil, even though the formulas on either side are only three point formulas. For $\nu = 1$ or

$\nu = 2$, i.e., using the rectangle or trapezoid rules, these formulas have clear finite volume interpretations, but for $\nu>2$ this is not the case. To show that any of these approximations give a weak solution if they converge, we write

$$\frac{v_j^{n+1}-v_j^n}{\Delta t} = \begin{cases} \dfrac{g(x,x+h)-g(x-h,x)}{h} & j\neq 0 \\ Q_0\, v_0^n & j=0 \end{cases}, \tag{3.4}$$

$Q$ as defined in (3.3). Multiply each side by $\phi_j^n$ and the weights $w_j$, and sum over all grid points,

$$\sum_j \frac{v_j^{n+1}-v_j^n}{\Delta t}\,\phi_j^n\, w_j h = \sum_{j\neq 0} \frac{g(v_j,v_{j+1})-g(v_{j-1},v_j))}{h}\,\phi_j^n w_j h + hQ_0 v_0 \phi_j^n w_0. \tag{3.5}$$

Applying summation by parts to the right hand side sum gives

$$\begin{aligned} \sum_{j\neq 0} &= \frac{g(v_j,v_{j+1})-g(v_{j-1},v_j)}{h}\phi_j^n w_j h \\ &= \sum_{|j|\geq 2} g(v_{j-1},v_j)(\phi_{j-1}^n w_{j-1}-\phi_j^n w_j) - g(v_0,v_1)\phi_1^n w_1 \\ &= \sum_{|j|\geq \nu+1} g(v_{j-1},v_j)(\phi_{j-1}^n-\phi_j^n) - g(v_0,v_1)\phi_1^n w_j \\ &\quad - \sum_{|j|=0}^{\nu} g(v_{j-1},v_j)(\phi_{j-1}^n w_{j-1}-\phi_j^n w_j). \end{aligned}$$

The first sum is already an $O(h)$ approximation to the integral. Since $\phi$ is smooth, we can expand $\phi_j$ in the remaining two terms around $\phi_0$, also with an error $O(h)$. By definition of $Q_0 v_0$, all the remaining terms multiplying $\phi_0$ disappear, to terms of $O(h)$, and we have

*Proposition 3.1: Any consistent approximation to the integral determines an interface equation that gives a weak solution to the pde, if the approximations converge as $h \to 0$ .*

We will use this to derive approximations in more complicated geometries by exactly conserving some $S$, rather than seeking a finite volume interpretation.

In general, it is difficult to say anything about the order of accuracy of the interface approximations so derived in the more common cases where $\nu=1$ or 2. For higher order integration approximations, however, we have the following.

Let

$$S^0 = I^0 + O(h^\nu) \tag{3.7}$$

where $I^0 = \int u^0\, dx$. The conservation law (1.1) exactly conserves $\int u\, dx$, so

$$I^0 = I^1$$

By design, the discrete approximation is conserved,

$$S^0 = S^1$$

and so

$$S^1 = I^1 + O(h^\nu). \tag{3.8}$$

Looking more closely at (3.8),

$$\begin{aligned} S^1 &= \sum_j h w_j u_j^1 \qquad (3.9)\\ &= \sum_{j\neq 0} h w_j (u(jh,\Delta t) + O(h^{p+1})) + h w_0 (u(0,\Delta t) + O(h^{\epsilon+1})), \end{aligned}$$

where we assume the numerical scheme on either side of the interface has order of accuracy $p$, (hence the local truncation error in the first step is of order $p+1$), and the interface approximation has unknown order of accuracy $\epsilon$. Continuing, we have

$$S^1 = \sum_j h w_j (u(jh,\Delta t) + O(h^{p+1}) + w_0\, O(h^{\epsilon+2}). \tag{3.10}$$

Equation (3.10) follows since the error term $O(h^{p+1})$ in the summation is multiplied by $h$, but summed over $1/h$ grid points.

The sum remaining in (3.10) is the same integration rule applied to the true solution $u$ at the first time step, and so

$$S^1 = I^1 + O(h^\nu) + O(h^{p+1}) + w_0 O(h^{\epsilon+2}). \tag{3.11}$$

Therefore, by (3.8) the last two terms in (3.11) must cancel up to terms of order $h^\nu$. Thus for $\nu \geq p+1$, $\epsilon \geq p-1$, showing

*Proposition 3.2: The interface approximation is not less than one order of accuracy less than the accuracy of the scheme itself, for an integration rule with order of accuracy $\nu \geq p+1$.*

Unfortunately, for $\nu \leq p$, this argument only shows that $\epsilon \geq \nu-2$. For example, if $\nu=2$ for the trapezoid rule, and $p=2$ for many numerical methods, we only have $\epsilon \geq 0$. In practise however, we have always obtained $\epsilon=1$, or in general $\epsilon = \min(p-1,\nu-1)$ for $\nu \leq p$ as well. If this weren't true, the step where $u_j^1 = u(jh,n\Delta t) + O(h^p)$ would not be true, since the global order of accuracy of the solution would drop to $\epsilon+1$, or one more than the order of accuracy of the boundary formula ( see Gustafsson, [1975]).

## 4 Overlapping Grids in One Dimension

In this section we apply the previous technique to derive interface conditions for a one dimensional model problem where two grids overlap, and do not share a common grid line. The configuration is shown in figure 4.1. Notice that we have switched to using cell centered rather than grid point centered variables in this section.

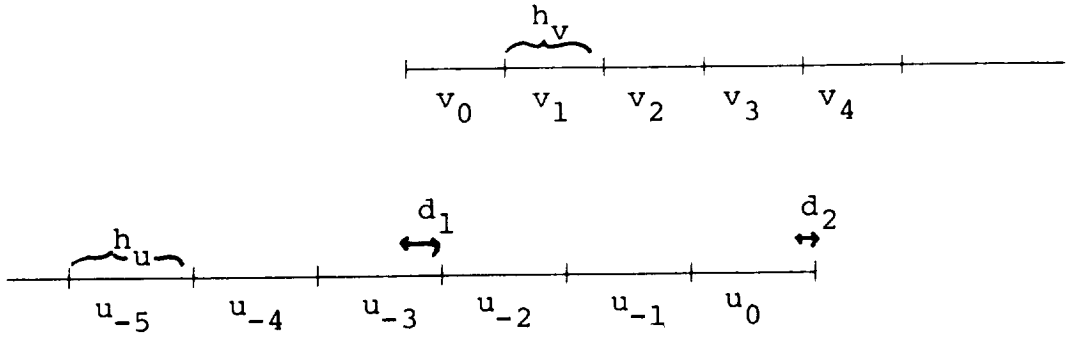


**Figure 4.1** Overlapping Grids in One Dimension

Interface conditions are needed for $u_0$ and $v_0$. Unfortunately, the simplest procedure of interpolating for $u_0^{n+1}$ at the new time level from the interior integration of $v$, and vice versa, is not conservative in any way.

Again, we start by defining a discrete approximation to the integral of a function on the entire domain. Outside of the overlap region, the integration rule we use will be the second order accurate midpoint rule. In the region of overlap both grids will contribute equally to the approximation of the integral. In particular, the cell around $u_{-3}$ will contribute $(h_u - d_1)\, u_{-3}$ to the part of the integral without overlapping, and $\frac{d_1}{2}\, u_{-3}$ to the overlapped region. Thus,

$$S = h_u \sum_{j \le -4} u_j + (h_u - d_1)\, u_{-3} + \left( \frac{d_1 u_{-3} + h_u \sum_{j=-2}^{0} u_j}{2} \right) \qquad (4.1)$$

$$+ \left( \frac{\sum_{j=0}^{3} h_v v_j + d_2 v_4}{2} \right) + (h_v - d_2) v_4 + \sum_{j \ge 5} h_v v_j .$$

This is still a second order accurate integration rule, since the approximation drops to first order at only two cells.

The point of figure 4.1, and the approximation 4.1, is that we are not combining the grids into one global grid, as is indicated in figure 4.2 for example. This would leave a tiny cell in the interior of the grid where special formulas would be needed to avoid instability from exceeding the CFL condition. (We say more about this possibility later) The point of figure 4.1 is that two calculations on regular grids will be done independently, one on each grid, and extra work will be done only at the boundary of each grid.

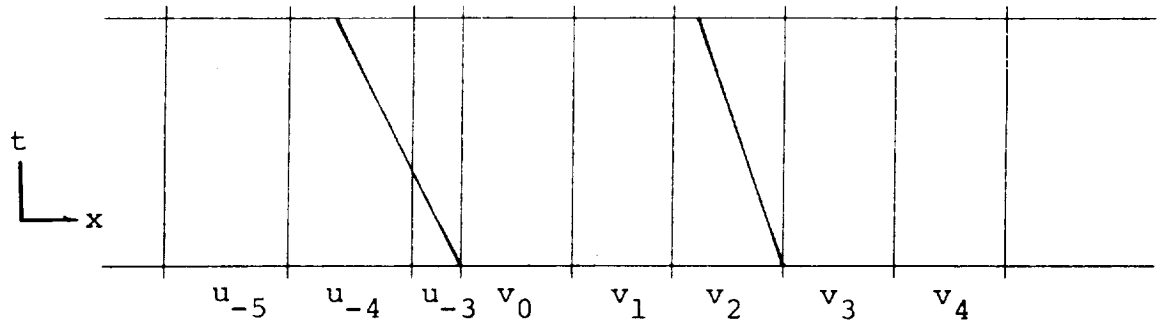


**Figure 4.2** Merging Overlapping Grids Into One Global Grid.

Using (4.1), we calculate the discrete flux change

$$\frac{S^1-S^0}{\Delta t} = g(u_{-3},u_{-2}) - \frac{d_1}{2h_j}(g(u_{-3},u_{-2}) - g(u_{-4},u_{-3})) \qquad (4.2)$$

$$+ \frac{1}{2}(-g(u_{-3},u_{-2}) + g(u_{-1},u_0)) + \frac{h_u}{2}Qu_0 + \frac{h_v}{2}Rv_0$$

$$+ \frac{1}{2}(-g(v_0,v_1)+g(v_3,v_4)) - \frac{d_2}{2h_v}(g(v_4,v_5)-g(v_3,v_4)) - g(v_3,v_4),$$

where $Qu_0$ is defined by $u_0^{n+1} = u_0^n + \Delta t Qu_0$, and similarly for $Rv_0$. As before, the fluxes that remain are from the neighboring cells where the weights $w_j$ in the integration rule change.

For conservation, we set $\frac{S^1-S^0}{\Delta t} = 0$, and divide the fluxes in (4.2) according to

which interface they came from. This gives the interface approximations

$$Qu_0 = \frac{\frac{d_2}{h_v}\, g(v_4,v_5) - (1-\frac{d_2}{h_v})\, g(v_3,v_4) - g(u_{-1},u_0)}{h_u} \tag{4.3}$$

$$Rv_0 = \frac{g(v_0,v_1) - \frac{d_1}{h_u}\, g(u_{-4},u_{-3}) - (1-\frac{d_1}{h_u})\, g(u_{-3},u_{-2})}{h_v}$$

Thus, the missing flux at the left end of the cell for $v_0$ is linearly interpolated from the fluxes on the other grid, and vice versa. This gives a first order accurate formula for $u_0$ and $v_0$. Notice that the interior boundaries are not treated as inflow or outflow boundaries, and that the formulas are independent of the direction of the characteristics at the interface. This information will appear in the definition of the numerical flux function $g$ itself.

The linear stability of the interpolation formulas for overlapping grids in 1D was proved by Starius [1980] using the Lax Wendroff scheme on both grids, and by Reyna [1982] and Berger [1983] for the leapfrog scheme. (If the equation is linear, then interpolation of the fluxes is equivalent to interpolation for the solution itself.) We point out that if the two grids meet at a common grid point without overlapping, then for certain ratios of the mesh spacing ($h_u = h_v, h_u = 2h_v$), the leapfrog scheme is unstable.

Since linear interpolation gives a first order accurate interface approximation, it is reasonable to wonder if higher order interpolation of the fluxes would yield a more accurate approximation. For example, suppose we define $Rv_0$ by cubic interpolation, and simplify notation by assuming the left end of the $v$ grid lies exactly in the middle of the cell for $u_{-3}$,(i.e. $d_1 = h_u/2$ ). Then

$$Rv_0 = \frac{-g(u_{-2},u_{-1}) + 9g(u_{-3},u_{-2}) + 9g(u_{-4},u_{-3}) - g(u_{-5},u_{-4})}{16}.$$

This corresponds to conserving an approximation

$$S = \sum_{j\geq 0} \frac{h_v}{2} v_j + \sum_{j\leq -5} h_u u_j + \sum_{j\geq -1} \frac{h_u}{2} u_j + \sum_{j=-4}^{-2} h_u w_j u_j,$$

where $w_{-2} = \frac{15}{24}$, $w_{-3} = \frac{24}{32}$, and $w_{-4} = \frac{33}{32}$. However, it is unlikely that this particular approximation is stable, since the interpolation coefficients are not positive.

Before moving on to the 2D case, we raise one other possibility that we have only considered in 1D. This idea is based on using the global grid of figure 4.2, and the large

time step methods of LeVeque [1982]. The problem with the grid in figure 4.2 is that the tiny cell associated with the value $u_{-3}$ will force a reduction in the time step to maintain stability, if no modification of the difference scheme is made. In the large time step algorithm of LeVeque, instead of calculating fluxes at the cell boundary, and then differencing the fluxes to update the cell value, the jump in the solution at the cell boundary is propagated, in the locally correct direction, and added to the existing value to get the solution at the next time step. This is what Roe calls *increment form*. With a CFL condition $> 1$, the differences can propagate several cells away, and are not limited to one or the other of the cells immediately adjacent to the boundary. For example, in figure 4.2, consider the equation $u_t + au_x = 0$, with $a<0$. The jump $v_3^n - v_2^n$ is propagated completely to the left, with a speed of propagation $a$ for time $\Delta t$, landing completely in the cell associated with $v_2$. The total amount of $v_2$ at the new time is

$$h_v v_2^{n+1} = h_v v_2^n + \Delta t a (v_3^n - v_2^n), \tag{4.4}$$

which is just upwind differencing. For a nonlinear equation, $u_t + f(u)_x = 0$, (4.4) is replaced by

$$h_v v_2^{n+1} = h_v v_2^n + \Delta t \left( \frac{f_{v_3} - f_{v_2}}{v_3 - v_2} \right) (v_3^n - v_2^n). \tag{4.5}$$

Now consider the smaller cell for $u_{-3}$ in figure 4.2. The jump $(v_0 - u_{-3})$ can propagate completely through $u_{-3}$'s cell, and partially into $u_{-4}$. The distance that the jump reaches into the cell of $u_{-4}$ is $(\frac{f_{v_0} - f_{u_{-3}}}{v_0 - u_{-3}})\Delta t - h_{-3}$, giving the difference equations

$$u_{-3}^{n+1} = u_{-3}^n + \left( \frac{f_{v_0} - f_{u_{-3}}}{v_0 - u_{-3}} \right) \frac{\Delta t}{h_{-3}} (v_0 - u_{-3})$$

$$u_{-4}^{n+1} = u_{-4}^n + \left( \frac{f_{u_{-3}} - f_{u_{-4}}}{u_{-3} - u_{-4}} \right) \frac{\Delta t}{h_u} (u_{-3} - u_{-4}) + \frac{\left[ \left( \frac{f_{v_0} - f_{u_{-3}}}{v_0 - u_{-3}} \right) \Delta t - h_{-3} \right]}{h_u} (v_0 - u_{-3})$$

Total conservation follows from the definition of the algorithm.

We have not yet considered the generalization of this algorithm to 2D and to rotated grids, where the direction of propagation of a jump is not clearly defined. A algorithm related to this, called flux redistribution, has been proposed by Chern and Colella [1984] in connection with a front tracking scheme.

## 5 Overlapping Grids in Two Dimensions

In this section we formulate a general algorithm to give conservative boundary equations for meshes which overlap in an arbitrary fashion in two dimensions. We start by first considering the very special case of 2 grids which are rotated by $\pi/4$ with respect to each other, and where the grid lines of one go through the corners of the other, as in figure 5.1. We will quickly go through several more complicated pictures to derive the more general algorithm. However, it is in the simpler cases that the physical interpretation of the results is clear.

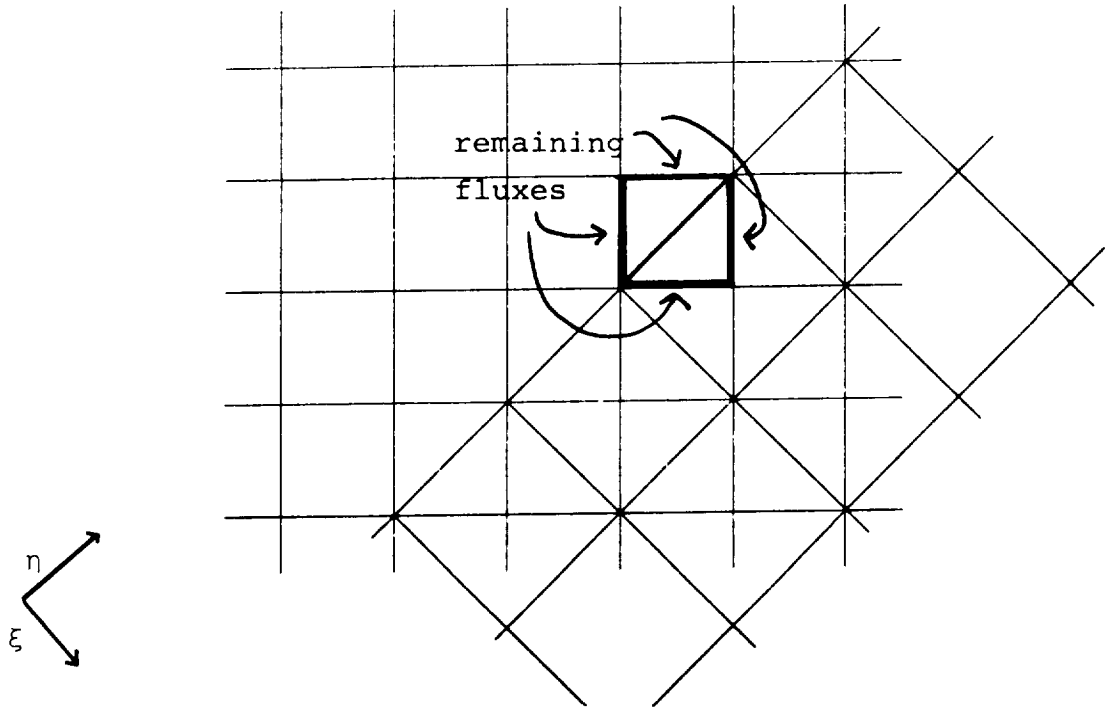


**Figure 5.1** Two grids rotated by $\pi/4$ with respect to each other.

As before, boundary conditions are needed only for the rotated grid across the line $\xi = 0$. We will assume the grids overlap by as many cells as necessary, which is typically 2 or 3, depending on the relative mesh sizes of the two grids. On the vertical grid, a time step can proceed completely ignoring the rotated grid, and ignoring boundary conditions for the vertical grid. Extra work will be done only at boundaries, and not in the interior of a grid. Otherwise, it would lead to complicated data structures, and a loss of vectorization. By keeping the interior computations separate, they can also be done in parallel.

We define the discrete conserved quantity $S$ using the solution on the vertical grid where it stands alone, and both grids with a weight of 1/2 where they overlap. The

diagonal cells in the vertical grid which intersect the boundary will have a weight $\frac{h^2}{2}$ from the non-overlapping triangular part of the cell, and $\frac{h^2}{4}$ from the overlap region. Being imprecise with notation to avoid a mess, we have

$$S = \sum_{\substack{\text{nonoverlapped} \\ \text{region}}} h^2 u_{ij} + \sum_{\text{diagonal}} \frac{3h^2}{4} u_{ii} + \sum_{\substack{\text{overlapped} \\ \text{region}}} \frac{h^2}{2} u_{ij} + \sum_{\substack{\text{rotated} \\ \text{grid}}} \frac{(\sqrt{2}h)^2}{2} v_{ij}. \quad (5.1)$$

The solution on the vertical grid is denoted by $u$, and on the rotated grid by $v$. We seek a definition for the missing flux $\hat{f}_j$ in the integration of $v_{1j}$,

$$v_{1j}^{n+1} = v_{1j}^{n} + \frac{\Delta t}{\Delta \eta}\left[\hat{g}(v_{1j}, v_{1j+1}) - \hat{g}(v_{1j-1}, v_{1j})\right] + \frac{\Delta t}{\Delta \xi}\left[\hat{f}(v_{1j}, v_{2j}) - \hat{f}_j\right].$$

In calculating $\frac{S^1 - S^0}{\Delta t}$, the terms that remain are from neighboring cells in the vertical grid where the integration weights in (5.1) change. These fluxes are marked with dark lines in figure 5.1. Clearly, by defining

$$\frac{\hat{f}_{1/2}}{\Delta \xi} = \frac{1}{4h}\left[f_{i-1/2,i} + f_{i+1/2,i}\right] - \frac{1}{4h}\left[g_{i,i-1/2} + g_{i,i+1/2}\right]$$

the fluxes balance. Since $\Delta \xi = \sqrt{2}\, h$ in figure 5.1, we get the formula

$$\hat{f}_j = \frac{1}{2\sqrt{2}}(f_{i-1/2,i} + f_{i+1/2,i}) - \frac{1}{2\sqrt{2}}(g_{i,i-1/2} + g_{i,i+1/2}). \quad (5.3)$$

On the vertical grid the equation being solved is $u_t + f(u)_x + g(u)_y = 0$. On a grid rotated by $\frac{\pi}{4}$, using the coordinates illustrated in figure 5.1, the equation becomes

$$u_t + \hat{f}_\xi + \hat{g}_\eta = 0, \quad (5.4)$$

where

$$\hat{f} = f\cos\theta + g\sin\theta$$

$$\hat{g} = f\cos\theta - g\sin\theta.$$

For the case $\theta = -\pi/4$, the flux function $\hat{f} = \frac{1}{\sqrt{2}}(f - g)$, and so (5.3) is linear interpolation for the correct fluxes in the new coordinate system, and so is second order accurate if the method on the vertical grid is second order.

Two more observations complete this case before considering the more general situation. First, consider what happens if the grid spacing on the rotated grid is halved. The fluxes that don't cancel on the vertical grid are the same, whereas two interface

equations are needed on the rotated grid. Everything balances by assigning

$$\hat{f}_j = \frac{1}{2\sqrt{2}}(f_{i-1/2,i} - g_{i,i-1/2})$$

$$\hat{f}_{j+1/2} = \frac{1}{2\sqrt{2}}(f_{i+1/2,i} - g_{i,i+1/2}),$$

The fluxes are separated depending on whether they lie above or below the line $\eta = j+1/2$.

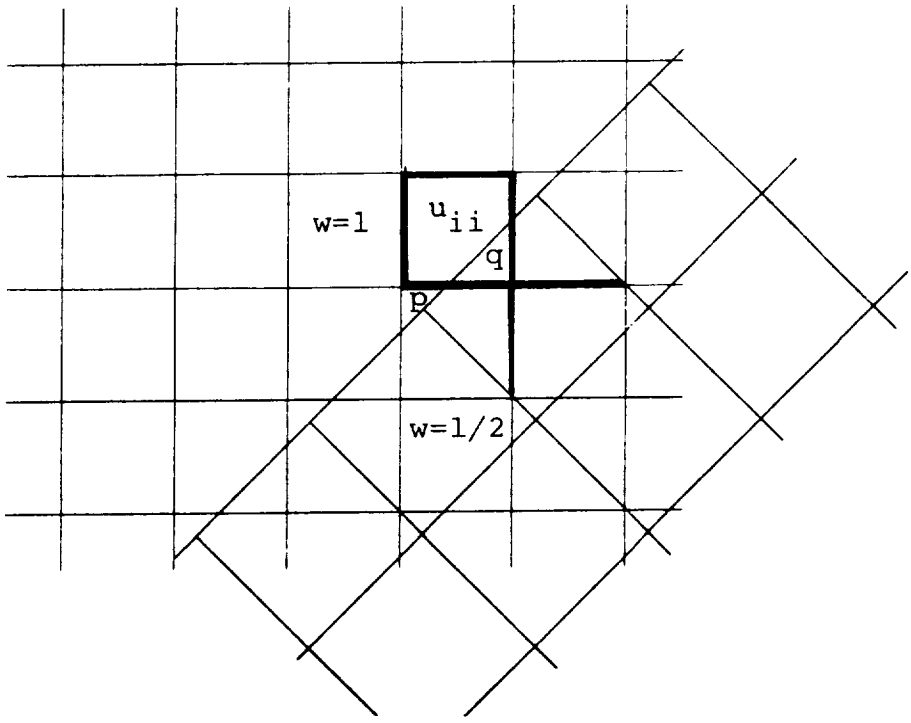


**Figure 5.2** Two grids rotated by $\pi/4$ with irregular overlap.

The second generalization is shown in figure 5.2. The grids are still rotated by $\pi/4$, but the grid lines of one do not exactly pass through the corners of the other grid. The weights used in the integration rule for the $u$ grid are marked in figure 5.2, where the overlapping cell areas are denoted by $p$ and $q$ as indicated. The vertical grid fluxes that determine the flux into the rotated grid are darkened in figure 5.2. The resulting interface equations are

$$\hat{f}_j = \frac{\sqrt{2}}{2}\Big\{q\,[f_{i-1/2,i} - g_{i,i+1/2}] + (1-p-q)\,[f_{i+1/2,i} - g_{i,i-1/2}] \qquad (5.5)$$

$$+ p[f_{i+1/2,i-1} - g_{i+1,i-1/2}]$$

Again, all the fluxes from the vertical grid lying between grid lines $j$ and $j+1$ are used to define the missing flux. This 6 point stencil is not equivalent to a finite volume formula in a natural way.

Leaping ahead to the general case, consider figure 5.3. The interface equation for $\hat{f}_j$ is a linear combination of the neighboring fluxes,

$$\hat{f}_{1/2,j} = \alpha f_{11} + \beta f_{12} + \gamma f_{21} + \delta f_{22} + \epsilon g_{11}, \qquad (5.6)$$

with coefficients determined by the amount of overlap and the integration rule. For example, to find the coefficient $\gamma$ for the contribution of $f_{12}$, first we need to know the differing weights of the adjacent cells in the integration rule. The cell to the left has weight $w_{0,2} = 1$. To the right the weight is determined by the amount of overlap, call it $p$, so the weight is $w_{1,2} = 1-\frac{p}{2}$. The flux contribution is for $f_{1,2}$ is then $w_{0,2}-w_{1,2}$ or $\frac{p}{2}$ in total. Some of this flux is also used to determine $\hat{f}_{1/2,j+1}$. The proportion given to $\hat{f}_j$ is determined by the ratio $\frac{l}{h}$, where $l$ is the length subtended by extending the coordinate line $\eta = j$.

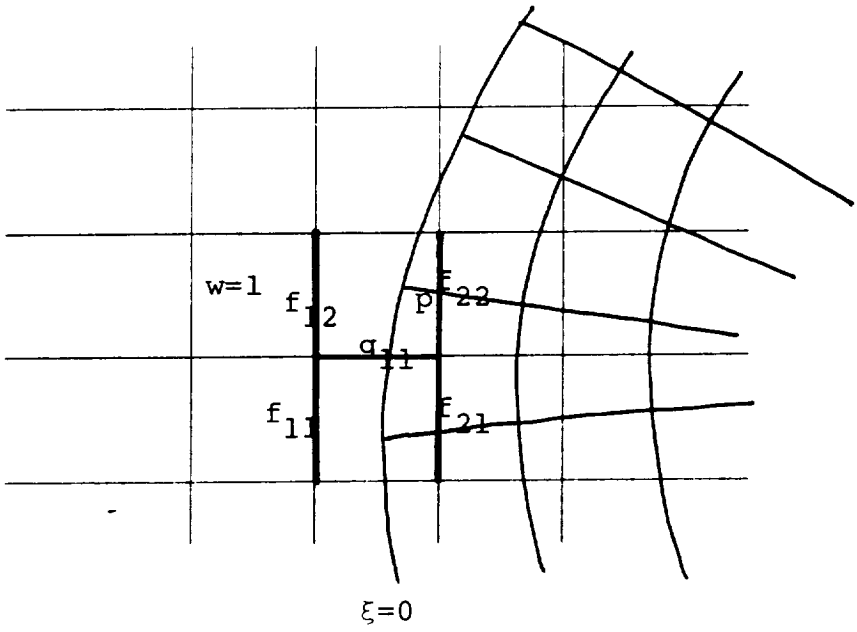


**Figure 5.3** General Overlapping Grids in 2D.

This interface procedure can be implemented for general 2D grids in a practical way with the following steps. When the initial gridding is done, some preprocessing is needed to determine

(a) the area of intersection for the cells in the main grid, and

(b) the intersection of the normal coordinate lines of the new grid with all cell sides of the main grid. If there is no such intersection, all of the flux on a side is used.

Step (a) determines the weight of the cells in the integration rule, and step (b) determines the amount of the main grid's flux to be apportioned to the $j^{th}$ boundary cell. Since two quadrilateral grids can intersect in a many sided polygon, depending on the mesh ratios of the grids, these steps can be complicated. In step (a), it is possible that the conservative interpolation procedure of Dukowicz [1984] for rezoning lagrangian calculations will be helpful in calculating overlap areas. In any case, this preprocessing is only done once. The coefficients for the linear combinations in the boundary equations are saved in a data structure corresponding to the boundary of the overlapping grid. During the time integration, we repeat step

(c) when the fluxes on the main grid are calculated, the appropriate amount of each flux is transmitted to the new grid.

This procedure does involve extra storage overhead proportional to the length of the grid boundary. We point out that it applies even if an operator split scheme is used, so that not all fluxes are available at the same time.

**Acknowledgments**

I am grateful to Jonathan Goodman for valuable discussions during the course of this work.

**References**

[1] E. Atta, *Component-Adaptive Grid Interfacing,* AIAA Paper No. 81-0382. Presented at the AIAA 19$^{th}$ Aerospace Sciences Meeting, 1981.

[2] J. A. Benek, J. L. Steger and F. C. Dougherty, *A Flexible Grid Embedding Technique with Application to the Euler Equations,* AIAA Paper No. 83-1944. Presented at the 6$^{th}$ Computational Fluid Dynamics Conference, Danvers, Mass. July, 1983.

[3] M. Berger, *Adaptive Mesh Refinement for Hyperbolic Partial Differential Equations,* PhD dissertation, Computer Science Dept., Stanford University, 1982.

[4] M. Berger, *Stability of Interfaces with Mesh Refinement*, ICASE Report No. 83-42, August, 1983. To appear in Math. Comp.

[5] M. Berger and A. Jameson, *Automatic Adaptive Grid Refinement for the Euler Equations*, MAE Report No. 1633, Princeton University, October, 1983. Also as NYU Report No. DOE/ER/03077-202. To appear in AIAA J.

[6] M. Berger and J. Oliger, *Adaptive Mesh Refinement for Hyperbolic Partial Differential Equations*, J. Comp. Phys. 53 (1984), 484-512.

[7] J. Bolstad, *An Adaptive Finite Difference Method for Hyperbolic Systems in One Space Dimension*, PhD dissertation, Computer Science Dept., Stanford Universty, 1982.

[8] C. Boppe, *Calculation of Transonic Wing Flows by Grid Embedding*, AIAA Paper No. 77-207.

[9] A. Brandt, *Guide to Multigrid Development*, Lecture Notes in Mathematics 960, Springer-Verlag, 1981.

[10] I. Chern and P. Collela, *A Conservative Front-Tracking Algorithm for Hyperbolic Conservation Laws*. In Preparation.

[11] J. Dukowicz, *Conservative Rezoning (Remapping) for General Quadrilateral Meshes*, J. Comp. Phys. 54 (1984), 411-424.

[12] W. Gropp, *A Test of Moving Mesh Refinement for 2D Scalar Hyperbolic Problems*, SIAM J. Sci. and Stat. Comp. 1 (1980), 191-197.

[13] B. Gustafsson, *The Convergence Rate for Difference Approximations to General Mixed Initial Value Problems*, Math. Comp. 29 (1975), 649-686.

[14] B. Gustafsson, H.-O. Kreiss, and A. Sundström, *Stability Theory of Difference Approximations for Initial Boundary Value Problems. II*, Math. Comp. 26 (1972), 649-686.

[15] K. Hessenius and T. Pulliam, *A Zonal Approach to Solution of the Euler Equations*, AIAA Paper No. 82-0969, St. Louis, Missouri, June 1982.

[16] P. Lax, *Hyperbolic Systems of Conservation Laws and the Mathematical Theory of Shock Waves*, SIAM, 1972.

[17] P. Lax and B. Wendroff, *Systems of Conservation Laws*, Comm. Pure and Appl. Math 13 (1960), 217-237.

[18] R. LeVeque, *Large Time Step Shock Capturing Techniques for Scalar Conservation Laws*, SIAM J. Numer. Anal.18 (1922), 1091-1109.

[19] S. McCormick, *The Fast Adaptive Composite (FAC) Method for Elliptic Equations,* Submitted to Math. Comp.

[20] M. Mock and P. Lax, *The Computation of Discontinuous Solutions of Linear Hyperbolic Equations,* Comm. Pure and Appl. Math 21 (1978), 423-430.

[21] S. Osher and R. Sanders, *Numerical Approximations to Nonlinear Conservation Laws with Locally Varying Time and Space Grids.* Preprint.

[22] M. Rai, *A Conservative Treatment of Zonal Boundaries for Euler Equations Calculations,* AIAA Paper No. 84-0164.

[23] L. Reyna, *On Composite Meshes,* Ph. D. dissertation, California Institute of Technology, 1982.

[24] G. Starius, *On Composite Mesh Difference Schemes for Hyperbolic Differential Equations,* Numer. Math. 35 (1980), 241-255.

[25] R. Warming and R. Beam, *Upwind Second-Order Difference Schemes and Applications in Aerodynamic Flows,* AIAA J. 14 (1976), 1241-1249.